K 6

```
EEEEEEEEEEEEEEE   RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE   RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE   RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE               RRR         RRR  FFF
EEE               RRR         RRR  FFF
EEE               RRR         RRR  FFF
EEE               RRR         RRR  FFF
EEE               RRR         RRR  FFF
EEE               RRR         RRR  FFF
EEEEEEEEEEEE      RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE      RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE      RRRRRRRRRRRR     FFFFFFFFFFFF
EEE               RRR   RRR        FFF
EEE               RRR   RRR        FFF
EEE               RRR   RRR        FFF
EEE               RRR      RRR     FFF
EEE               RRR      RRR     FFF
EEE               RRR      RRR     FFF
EEEEEEEEEEEEEEE   RRR         RRR  FFF
EEEEEEEEEEEEEEE   RRR         RRR  FFF
EEEEEEEEEEEEEEE   RRR         RRR  FFF
```

```
_$

Va
--
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
```

K 14

```
**FILE**ID**UBAINT
```

```
UU      UU  BBBBBBBB      AAAAAA      IIIIII    NN      NN  TTTTTTTTTT
UU      UU  BBBBBBBB      AAAAAA      IIIIII    NN      NN  TTTTTTTTTT
UU      UU  BB      BB  AA      AA      II      NN      NN      TT
UU      UU  BB      BB  AA      AA      II      NN      NN      TT
UU      UU  BB      BB  AA      AA      II      NNNN    NN      TT
UU      UU  BB      BB  AA      AA      II      NNNN    NN      TT
UU      UU  BBBBBBBB    AA      AA      II      NN  NN  NN      TT
UU      UU  BBBBBBBB    AA      AA      II      NN  NN  NN      TT
UU      UU  BB      BB  AAAAAAAAAA      II      NN    NNNN      TT
UU      UU  BB      BB  AAAAAAAAAA      II      NN    NNNN      TT
UU      UU  BB      BB  AA      AA      II      NN      NN      TT          ....
UU      UU  BB      BB  AA      AA      II      NN      NN      TT          ....
UUUUUUUUUU  BBBBBBBB    AA      AA    IIIIII    NN      NN      TT          ....
UUUUUUUUUU  BBBBBBBB    AA      AA    IIIIII    NN      NN      TT          ....

LL              IIIIII        SSSSSSSS
LL              IIIIII        SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

UN

AR

LA

FU

CO

CO

```
C
C Version:      'V04-000'
C
C****************************************************************************
C*                                                                          *
C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
C*  ALL RIGHTS RESERVED.                                                    *
C*                                                                          *
C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
C*  TRANSFERRED.                                                            *
C*                                                                          *
C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
C*  CORPORATION.                                                            *
C*                                                                          *
C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
C*                                                                          *
C*                                                                          *
C****************************************************************************
C
C
C
C       AUTHOR  BRIAN PORTER            CREATION DATE   3-FEB-1979
C
C       Functional description:
C
C       This module is called to display UBA interrupts.
C
C       Modified by:
C
C       V03-004 SAR0160         Sharon A. Reynolds,     13-Oct-1983
C               Added an SYE update that makes the register heralds
C               generic.
C
C       V03-003 SAR0103         Sharon A. Reynolds,     20-Jun-1983
C               Changed the carriage control in the 'format' statements
C               for use with ERF.
C
C       V03-002 SAR0054         Sharon A. Reynolds,     13-Jun-1983
C               Removed brief/cryptic support.
C
C       v03-001 BP0001          Brian Porter,           20-AUG-1982
C               Minor edit.
C**

        Subroutine UBA_INTERRUPTS (lun,option)

        include 'src$:msghdr.for /nolist'


        byte            lun
```

```

                character*1     option



                if (
                1 lib$extzv(24,8,emb$l_hd_sid) .eq. 255         ! 11/780
                1 .or.
                1 lib$extzv(24,8,emb$l_hd_sid) .eq. 1           ! 11/780's
                1 .or.
                1 lib$extzv(24,8,emb$l_hd_sid) .eq. 4           ! 11/7XX
                1 ) then

                if (option .eq. 'S') then

                call dw780_interrupt (lun)
                endif

        c
        c       for additional UBA support the ELSE-IF-THEN should be expanded
        c       at this point.
        c

                endif

                return
                end
```

```
PROGRAM SECTIONS

    Name                                    Bytes   Attributes

  0 $CODE                                      80   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                      8   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                     32   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 EMB                                       512   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                     632

ENTRY POINTS

    Address  Type  Name

  0-00000000        UBA_INTERRUPTS

VARIABLES

    Address  Type  Name                                    Address  Type  Name

  3-00000000  I*4  EMB$L_HD_SID                          3-00000004  I*2  EMB$W_HD_ENTRY
  3-0000000E  I*2  EMB$W_HD_ERRSEQ                      AP-00000004@ L*1  LUN
 AP-00000008@ CHAR OPTION
```

```
ARRAYS

    Address  Type  Name                       Bytes  Dimensions

  3-00000000  L*1   EMB                          512  (0:511)
  3-00000006  I*4   EMB$Q_HD_TIME                  8  (2)

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                                Type  Name

        DW780_INTERRUPT                      I*4  LIB$EXTZV



        SUBROUTINE DW780_INTERRUPT (LUN)


        INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'



        BYTE            LUN

        INTEGER*4       FIELD
        INTEGER*4       COMPRESS4
        INTEGER*4       UBA_CSR
        INTEGER*4       UBA_CR
        INTEGER*4       UBA_SR
        INTEGER*4       UBA_DCR
        INTEGER*4       UBA_FMER
        INTEGER*4       UBA_FUBAR
        INTEGER*4       ADAPTER_TR
        INTEGER*4       ERROR_PC
        INTEGER*4       ERROR_PSL

        logical*1       diagnostic_mode

        EQUIVALENCE     (EMB(16),UBA_CSR)
        EQUIVALENCE     (EMB(20),UBA_CR)
        EQUIVALENCE     (EMB(24),UBA_SR)
        EQUIVALENCE     (EMB(28),UBA_DCR)
        EQUIVALENCE     (EMB(32),UBA_FMER)
        EQUIVALENCE     (EMB(36),UBA_FUBAR)
        EQUIVALENCE     (EMB(40),ADAPTER_TR)
        EQUIVALENCE     (EMB(44),ERROR_PC)
        EQUIVALENCE     (EMB(48),ERROR_PSL)

        CHARACTER*31    V1UBACR(0:6)
        DATA    V1UBACR(0)      /'ADAPTER INIT*'/
        DATA    V1UBACR(1)      /'UNIBUS POWER FAIL*'/
        DATA    V1UBACR(2)      /'CONFIGURATION INTERRUPT ENABLE*'/
        DATA    V1UBACR(3)      /'SBI TO UNIBUS ERROR INT ENABLE*'/
        DATA    V1UBACR(4)      /'UNIBUS TO SBI ERROR INT ENABLE*'/
        DATA    V1UBACR(5)      /'BR INTERRUPT ENABLE*'/
        DATA    V1UBACR(6)      /'INTERRUPT FIELD SWITCH*'/

        CHARACTER*25    V1UBASR(0:10)
        DATA    V1UBASR(0)      /'UNIBUS ''SSYN'' TIMEOUT*'/
        DATA    V1UBASR(1)      /'UNIBUS SELECT TIMEOUT*'/
        DATA    V1UBASR(2)      /'LOST ERROR*'/
        DATA    V1UBASR(3)      /'MAP REGISTER PARITY FAIL*'/
        DATA    V1UBASR(4)      /'INVALID MAP REGISTER*'/
        DATA    V1UBASR(5)      /'DATA PATH PARITY ERROR*'/
        DATA    V1UBASR(6)      /'COMMAND TRANSMIT TIMEOUT*'/
        DATA    V1UBASR(7)      /'COMMAND TRANSMIT ERROR*'/
        DATA    V1UBASR(8)      /'CORRECTED READ DATA*'/
        DATA    V1UBASR(9)      /'READ DATA SUBSTITUTE*'/
        DATA    V1UBASR(10)     /'READ DATA TIMEOUT*'/

        CHARACTER*3     TR
        DATA    TR              /'TR '/


        CALL FRCTOF (LUN)

        call header (lun)

        call logger (lun,'''DW'' INTERRUPT')

        diagnostic_mode = .false.

        if (lib$extzv(28,3,uba_dcr) .ne. 0) diagnostic_mode = .true.

        CALL LINCHK (LUN,2)

        WRITE(LUN,10) ERROR_PC
10      FORMAT(/' ',T8,'ERROR PC',T24,Z8.8)

        if (.not. diagnostic_mode) then

        CALL VAXPSL (LUN,ERROR_PSL)

        CALL LINCHK (LUN,2)

        WRITE(LUN,12) ADAPTER_TR
12      FORMAT(/' ',T8,'ADAPTER TR# ',I<COMPRESS4 (ADAPTER_TR)>,'.')

        CALL UBA_REGA (LUN,UBA_CSR)
        else

        call linchk (lun,6)

        write(lun,13) error_psl,adapter_tr,uba_csr
13      format(/' ',t8,'ERROR PSL',t24,z8.8,//,
        1 t8,'ADAPTER TR# ',i<compress4 (adapter_tr)>,'.',//,
        1 t8,'''DW'' CSR',t24,z8.8)
        endif

        CALL LINCHK (LUN,1)

        WRITE(LUN,15) UBA_CR
15      FORMAT(' ',T8,'''DW'' CR',T24,Z8.8)

        if (.not. diagnostic_mode) then

        CALL OUTPUT (LUN,UBA_CR,V1UBACR,0,0,6,'0')

        FIELD = LIB$EXTZV(26,5,UBA_CR)

        IF (FIELD .NE. 0) THEN

        FIELD = (FIELD*16) - 1

        CALL LINCHK (LUN,1)

```

```
                WRITE(LUN,20) FIELD
        20      FORMAT(' ',T40,'MAPS 0. THRU ',I<COMPRESS4 (FIELD)>,'. DISABLED')
                ENDIF
                endif

                CALL LINCHK (LUN,1)

                WRITE(LUN,25) UBA_SR
        25      FORMAT(' ',T8,''DW'' SR',T24,Z8.8)

                if (.not. diagnostic_mode) then

                CALL OUTPUT (LUN,UBA_SR,V1UBASR,0,0,10,'0')

                DO 35,I = 24,27

                IF (JIAND(UBA_SR,2**I) .NE. 0) THEN

                CALL LINCHK (LUN,1)

                WRITE(LUN,30) I - 20
        30      FORMAT(' ',T40,'BRRVR ',I1,'. FULL')
                ENDIF

        35      CONTINUE
                endif

                CALL LINCHK (LUN,1)

                WRITE(LUN,40) UBA_DCR
        40      FORMAT(' ',T8,''DW'' DCR',T24,Z8.8)

                if (diagnostic_mode) then

                CALL LINCHK (LUN,1)

                WRITE(LUN,45)
        45      FORMAT(' ',T40,'DIAGNOSTIC MODE')
                endif

                CALL LINCHK (LUN,1)

                WRITE(LUN,50) UBA_FMER
        50      FORMAT(' ',T8,''DW'' FMER',T24,Z8.8)

                if (.not. diagnostic_mode) then

                IF (JIAND(UBA_SR,'000006F8'X) .NE. 0) THEN

                FIELD = LIB$EXTZV(0,9,UBA_FMER)

                CALL LINCHK (LUN,1)

                WRITE(LUN,55) FIELD
        55      FORMAT(' ',T40,'SELECTED MAP = ',I<COMPRESS4 (FIELD)>,'.')
                ENDIF
                endif
```

```

              CALL LINCHK (LUN,1)

              WRITE(LUN,60) UBA_FUBAR
      60      FORMAT(' ',T8,''DW'' FUBAR',T24,Z8.8)

              if (.not. diagnostic_mode) then

              IF (JIAND(UBA_SR,'00000003'X) .NE. 0) THEN

              FIELD = JISHFT(JIAND(UBA_FUBAR,'0000FFFE'X),-1)

              CALL LINCHK (LUN,1)

              WRITE(LUN,65) FIELD
      65      FORMAT(' ',T40,'UNIBUS ADDRESS = ',O5.5,'X (OCTAL)')
              ENDIF
              endif

              RETURN
              END
```

```
PROGRAM SECTIONS

    Name                                            Bytes   Attributes

  0 $CODE                                             941   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                            441   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                            792   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 EMB                                               512   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                            2686

ENTRY POINTS

    Address  Type  Name

  0-00000000       DW780_INTERRUPT

VARIABLES

    Address  Type  Name                                     Address  Type  Name

  3-00000028  I*4  ADAPTER_TR                             2-000001EC  L*1  DIAGNOSTIC_MODE
  3-00000000  I*4  EMB$L_HD_SID                           3-00000004  I*2  EMB$W_HD_ENTRY
  3-0000000E  I*2  EMB$W_HD_ERRSEQ                        3-0000002C  I*4  ERROR_PC
  3-00000030  I*4  ERROR_PSL                              2-000001F0  I*4  FIELD
  2-000001F4  I*4  I                                     AP-00000004@ L*1  LUN
  2-000001ED  CHAR TR                                     3-00000014  I*4  UBA_CR
  3-00000010  I*4  UBA_CSR                                3-0000001C  I*4  UBA_DCR
  3-00000020  I*4  UBA_FMER                               3-00000024  I*4  UBA_FUBAR
  3-00000018  I*4  UBA_SR
```

```
ARRAYS

   Address   Type  Name                     Bytes  Dimensions

 3-00000000  L*1   EMB                        512  (0:511)
 3-00000006  I*4   EMB$Q_HD_TIME                8  (2)
 2-00000000  CHAR  V1UBACR                    217  (0:6)
 2-000000D9  CHAR  V1UBASR                    275  (0:10)

LABELS

   Address    Label        Address   Label        Address   Label         Address   Label         Address   Label         Address   Label

 1-00000039  10'      1-0000004F  12'       1-0000006D  13'       1-000000B2  15'       1-000000C6  20'       1-000000ED  25'
 1-00000101  30'              **  35        1-00000119  40'       1-0000012E  45'       1-00000145  50'       1-0000015B  55'
 1-0000017B  60'      1-00000192  65'

FUNCTIONS AND SUBROUTINES REFERENCED

 Type  Name             Type  Name             Type  Name             Type  Name             Type  Name             Type  Name

  I*4  COMPRESS4              FRCTOF                 HEADER           I*4  LIB$EXTZV              LINCHK                 LOGGER
       OUTPUT                 UBA_REGA               VAXPSL

COMMAND QUALIFIERS

  FORTRAN /LIS=LIS$:UBAINT/OBJ=OBJ$:UBAINT MSRC$:UBAINT

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
  /F77  /NOG_FLOATING  /I4  /OPTIMIZE  /WARNINGS  /NOD_LINES  /NOCROSS_REFERENCE  /NOMACHINE_CODE  /CONTINUATIONS=19

COMPILATION STATISTICS

  Run Time:             4.00 seconds
  Elapsed Time:         10.42 seconds
  Page Faults:          181
  Dynamic Memory:       185 pages
```

0154 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY